

University of Dundee

Automating "human-like" example-use in mathematics

Pease, Alison; Martin, Ursula

Published in:
CEUR Workshop Proceedings

Publication date:
2018

Document Version
Peer reviewed version

[Link to publication in Discovery Research Portal](#)

Citation for published version (APA):

Pease, A., & Martin, U. (2018). Automating "human-like" example-use in mathematics. *CEUR Workshop Proceedings*, 2261, 6-15. <http://ceur-ws.org/Vol-2261/paper2.pdf>

General rights

Copyright and moral rights for the publications made accessible in Discovery Research Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from Discovery Research Portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Automating “human-like” example-use in mathematics

Alison Pease¹, Ursula Martin²

¹ University of Dundee

² University of Oxford

A.Pease@dundee.ac.uk, Ursula.Martin@cs.ox.ac.uk

Abstract

We describe two studies into ways in which human mathematicians use mathematical examples in their research. In the first study we bring together theoretical and empirical approaches to studying ways in which examples are used in mathematical research, concluding that examples are used for conjecture invention, understanding, plausibility-testing, disproof and modification. Where possible we describe corresponding efforts in automating these aspects of reasoning. In our second study we present an investigation based on grounded theory into example-use during an online mathematical conversation. These studies suggest ways in which “human-like” example-use in mathematics could be further automated.

1 Introduction

At an event that we organised a few years ago [1], leading mathematicians flagged the importance of collaborative systems that “think like a mathematician”, handle unstructured approaches such as the use of “sloppy” natural language and the exchange of informal knowledge and intuition not recorded in papers, and engage diverse researchers in creative problem solving. This accords with work of cognitive scientists, sociologists, philosophers and the narrative accounts of mathematicians themselves, which highlight the paradoxical nature of mathematical practice — while the goal of mathematics is to discover mathematical truths justified by rigorous argument, mathematical discovery involves “soft” aspects such as creativity, informal argument, error and analogy.

We followed the above event with an empirical study of what mathematicians talk about [22], and found that examples form the biggest single category. These may be examples of a concept, such as the set of natural numbers being an example of a group, and the numbers 3, 4, and 5 an example of a Pythagorean triple, or supporting or counterexamples to a conjecture, such as 2 and

3 being supporting examples of the conjecture that all integers have an even number of divisors, and 4 being a counterexample. The study found that examples are used for different reasons at different points in a conversation, for instance to understand a conjecture, to test it, or extend it. For instance, consider the following conversation, taken from an online forum for solving a conjecture [3]:

“If the points form a convex polygon, it is easy.”
[Anonymous July 19, 2011 @ 8:08 pm]

“Yes. Can we do it if there is a single point not on the convex hull of the points?” [Thomas H July 19, 2011 @ 8:09 pm]

“Say there are four points: an equilateral triangle, and then one point in the center of the triangle. No three points are collinear. It seems to me that the windmill can not use the center point more than once! As soon as it hits one of the corner points, it will cycle indefinitely through the corners and never return to the center point. I must be missing something here...” [Jerzy July 19, 2011 @ 8:17 pm]

“This isn’t true it will alternate between the centre and each vertex of the triangle.” [Joe July 19, 2011 @ 8:21 pm]

Here we see people raising simple examples to understand a conjecture, and proposing and discussing other examples in order to explore and test the conjecture.

In this paper we bring together work on example-use in mathematics research and relate it to automated reasoning. In the first study we consider a course-grained empirical study (§2.1) and review theoretical ideas on example-use (§2.2), describing resulting roles for examples and corresponding automated systems (§2.3). In the second we conduct a fine-grained analysis of a mathematical research conversation (§3) and show the importance of context, language and social pleasantries for talking about different kinds of example. We conclude by considering next steps in building collaborative systems for contributing examples in mathematical research (§4).

1.1 Motivation

The simulation of mathematical reasoning has been a driving force throughout the history of Artificial Intelligence research, yet despite significant successes (eg [10; 9; 13]) it has not achieved widespread adoption by mathematicians. An oft-cited reason for this is that current systems cannot do mathematics in the way that humans do: machine proofs are thought to be unclear, uninspiring and untrustworthy, as opposed to human proofs which can be deep, elegant and explanatory. Traditionally there have been two barriers to developing systems which produce “human-like” mathematics: firstly, it is difficult to know what this is; and secondly, it is difficult to automate [6; 12]. Recent developments in online collaborative mathematics, such as the *Polymath* and *MathOverflow* projects [4; 5] provide a remarkable opportunity for overcoming the first barrier: by providing a working record of the backstage of mathematics, complete with mistakes, deadends and unfinished work, these constitute a rich evidence base for understanding live mathematical practice. At the same time, we believe that we can start to overcome the second barrier by focusing in on a specific and particularly prominent aspect of mathematical practice – example-use in mathematics. This will allow us to (a) build on the increasing sophistication of model generators in automated reasoning, and (b) formulate constrained, measurable and achievable goals for automated “human-like” example-use in mathematics.

2 Study 1: Theories of example-use in mathematical research

2.1 An empirical study

The mathematical community have developed systems for “crowdsourcing” (albeit among a highly specialised crowd) the production of mathematics through collaboration and sharing [20]. The *Polymath* and *MiniPolymath* collaborative proofs, a new idea led by Gowers and Tao, are of particular interest to us: these use a blog and wiki for collaboration among mathematicians from different backgrounds and have led to major advances [11]. Also of interest are discussion fora which allow rapid informal interaction and problem solving; in seven years the community question answering system for research mathematicians *MathOverflow* has around 70,000 users and has hosted over 90,000 conversations. These systems provide substantial and unprecedented evidence for studying mathematical practice, allowing the augmentation of traditional ethnography with a variety of empirical techniques for analysing the texts and network structures of the interactions.

In an ethnographic analysis of such a record [23], we classified each conversation turn as relating to different aspects of mathematical activity and found that the largest single category was *examples*. Here we mean examples of

concepts or conjectures: mathematical objects, such as specific numbers, graphs or groups, which have particular properties. For instance, the number 2 is an example of the concept *prime number*, a supporting example of the *fundamental theorem of arithmetic*, and a counterexample to the conjecture that *all primes are odd*. Examples were used for different purposes at different stages of the discussion. One of the first comments was a simple supporting example of the conjecture – the only example explicitly raised in this context. Other supporting examples were raised as elaboration or as highlighting the necessity of a condition in order to explore the condition. One comment contained an argument as to why a particular example could not exist. Some examples – both support and counterexamples – were raised as clarification. 83% of the comments we categorised as concerning examples were about counterexamples (or examples of undetermined status).

In a study of a sample *mathoverflow* conversations we found that in a third of the responses explicit examples were given, as evidence for, or counterexamples to, conjectures. Requests for examples of a phenomenon or an object with particular properties also featured in our breakdown of questions as one of three predominant kinds [16]. These analyses suggest that examples play a fundamental role in mathematical collaboration – a conservative estimate puts it at a third of all mathematical conversations that we analysed.

2.2 Theoretical work

The small amount of theoretical work on the use of examples in mathematical research – itself based on real-world case studies – supports our findings. Polya wrote extensively about the value of examples in conjecture generation and testing [24], while Lakatos followed up Polya’s ideas with a demonstration of the role that counterexamples play, driving negotiation and development of concepts, conjectures and proofs [15].

In his “Induction in solid geometry” [25, Part III, pp 35–] Polya details how examples are invoked at different points to suggest, evaluate, develop and prove the Descartes-Euler conjecture that for any polyhedra, the number of vertices (V) minus the number of edges (E) plus the number of faces (F) is equal to 2. He starts with simple examples of polyhedra to find regularities and *formulating initial conjectures*. Once a conjecture has been found, *plausibility testing* is performed with examples getting gradually more complex, looking for examples which support rather than refute the conjecture. If it stands up to these then more severe examples are looked for – with the focus going from *finding supporting examples* to *looking for counterexamples*, leading to “very difficult” cases. If a conjecture survives this test, then a proof is sought. Lakatos takes over at this point, describing a rational reconstruction in which counterexamples drive conjecture and concept change and are used to strengthen a proof via var-

ious responses. These include three main methods of theorem formation – all triggered by counterexamples: monster-barring (concerned with concept development); exception-barring (concerned with conjecture development), and the method of proofs and refutations (concerned with proof development).

2.3 Examples in Automated Reasoning

Example, or model, generation is one of the successes of formal verification, interactive theorem proving, and automated reasoning, with substantial research underlying these achievements. Techniques underlying such systems include methods based on first order reasoning, constraint satisfaction, and propositional logic: see [27] for an overview.

We summarise our empirical studies in §2.1 and theoretical work presented in §2.2 as using examples for the following purposes: (i) conjecture invention; (ii) understanding a conjecture; (iii) plausibility-testing; (iv) disproving or modifying a conjecture. Simulations of (i) and (iv) can be found in automated reasoning: to the best of our knowledge there is little or no automated work on (ii) or (iii). The first purpose is scientific induction, which underlies the machine learning paradigm. Colton’s theory formation system HR [7] also uses examples, or objects of interest (such as specific groups or numbers), to drive theory development. The system uses production rules to form new concepts from old ones; measures of interestingness to drive a heuristic search; empirical pattern-based conjecture making techniques to find relationships between concepts, and third party logic systems such as the Otter theorem prover [19], the Mace model generator [17]. Otter is a first order resolution theorem prover which has been used for many discovery tasks in algebraic domains, e.g., [18]. Mace is a model generator which employs the Davis-Putnam method for generating models to first order sentences. The IsaCosy system by Johansson et al. is another instance of example-driven theory formation software [14] which performs inductive theory formation by synthesising conjectures from the available constants and free variables.

Along with colleagues, Pease has developed two software systems based on the fourth purpose above. The Theorem Modifier system (TM) [8] uses Lakatos’s exception-barring methods to provide a flexible automated theorem-proving system. This is able to take in a conjecture, try to prove it and if unsuccessful (either because the conjecture is too hard to prove or because it is false), use supporting or counter-examples to produce modified versions of the conjecture which it *can* prove. For instance, given the non-theorem that all groups are Abelian, TM states that it cannot prove the original result, but it has discovered that all *self-inverse* groups are Abelian. The HRL system [21] is a computational representation of Lakatos’s theory, in which all of his main methods for theory development are implemented. In

keeping with the dialectical aspect described by Lakatos, HRL is implemented in an agent architecture. Each agent has a copy of the HR system, and starts with a different database of examples to work with, and different interestingness measures. Agents send conjectures, concepts, counterexamples, or requests to a central agent, who choreographs a discussion, using the example-based methods prescribed by Lakatos to modify faulty conjectures.

3 Study 2: A fine-grained study of example-use in mathematics research

3.1 Source material

To apply empirical methods to the study of mathematical explanation we looked for a suitable source of data which, ideally, would capture the live production of mathematics rather than the finished outcome in textbook or journal paper; would exhibit examples in practice through capturing mathematical collaboration; and could be argued to represent the activities of a reasonable subset of the mathematical community. The dataset we chose was the first Mini-Polymath project, in 2009, an online collaboration on a blog to solve problems drawn from International Mathematical Olympiads.

The first Mini-Polymath project started on July 20th, 2009 @ 6:02 am and ended August 15th, 2010 @ 3:30 pm. The problem statement came from Problem 6 of the Math Olympiad that year:

Let a_1, a_2, \dots, a_n be distinct positive integers and let M be a set of $n - 1$ positive integers not containing $s = a_1 + a_2 + \dots + a_n$. A grasshopper is to jump along the real axis, starting at the point 0 and making n jumps to the right with lengths a_1, a_2, \dots, a_n in some order. Prove that the order can be chosen in such a way that the grasshopper never lands on any point in M .

The solution was found on 21st July, 2009 @ 11:16 am after 201 comments. The total conversation contained 356 comments and 32,430 words, and there were between 81 and 100 participants (some participants were anonymous). We analysed the first 160 comments, which is 80% of the conversation leading up to the solution.¹

3.2 Method

We used an approach based on grounded theory [26] to conduct a fine-grained study. This is a data-driven method to systematically build integrated sets of concepts in a topic where little is known. Researchers keep an open mind in order to build a theory which is purely

¹The reason for 80% rather than the full 100% was purely pragmatic: such close analysis is extremely time-consuming to perform and it was felt that results were sufficiently robust after 80%.

grounded in data rather than influenced by prior work. As is the standard in grounded theory, we followed four stages:

1. Open coding: Use the raw data to suggest code definitions (anchors that help to identify key points in the data).
2. Axial coding: Development of concepts by combining codes into collections of similar content.
3. Selective coding: Grouping the concepts into categories - put the data back together by making connections across codes, categories, and concepts.
4. Theory building: Compare the central phenomenon across several dimensions, and formulate the major themes which have emerged.

Here codes, concepts and categories are different levels of abstraction and are the building blocks for a grounded theory.

We used a software tool for grounded theory and mixed methods research, dedoose [2], shown in Figure 1. Here we see the codification of a comment. It can be seen that we sometimes applied multiple codes within a single comment, or sentence; even applying overlapping codes if appropriate. In this example for instance, the words “ugle”, “elusive” and “hope” (which occurs twice) are coded as *emotion or value words*, which has been categorised under *language*.

The coding was mainly done by the first author, who has a first degree in Mathematics, a PhD in a related discipline and more than 10 years experience studying mathematical reasoning. Her analysis was discussed at length and during different stages of analysis with the second author, who has a PhD in Mathematics and over 10 years experience as a professional research mathematician. Any differences of opinion were resolved, allowing us to align our ideas.

3.3 Findings

In order to build a substantive theory of example-use in mathematical research, we followed the stages of grounded theory as described below.

Open coding

In the first stage, we identified and coded code definitions of interest via open coding. In the example below, a participant gives a local conjecture about the shortest route and points out a problem example straightaway. This comment excerpt is coded as “Counterexample to a local conjecture”

I had hoped that the shortest would always be (m_k, m_{k+1}) , but this doesn't seem to be true: consider $A = (9, 10, 11, 30)$ and $M = (11, 30, 49)$.

The open coding process created a total of 98 loosely connected codes and descriptions. Since we coded the conversation into any appropriate size chunk of data and allowed multiple codings, we could get more code applications than number of comments: in fact we got significantly more, with 646 applications of the codes across our 160 comments.

Axial coding

In the second stage, we identified interrelationships between our codes and formed concepts by combining codes, to describe repeated patterns of interactions and problem solving strategies in the conversation. We found 23 concepts, including examples, and also error, explanation, question, re-representation, references to other mathematics, metaphors and requests for help (we show interrelationships for our examples category in Figure 2).

Selective coding/Theory building

In the third and fourth stages, we grouped concepts into categories, making connections across codes, categories, and concepts. The following main categories emerged: *context* (often a mathematical object such as a conjecture or a concept), the *language* used (for instance, emotive or values language, or metaphors), and the *social pleasantries* needed to keep the conversation flowing (for instance, thanking a participant for their contribution or introducing a new comment in a very humble way).

We show a visual representation of how these categories relate to each other and to different kinds of examples in Figure 3.

Context:

Hypothesized cases to explore a conjecture were very common, and these arose in the context of introducing notation, re-representation, meta-comments about proof, explanation, justification, induction and using them to asking questions. Simple supporting examples were used to explore a conjecture.

Counterexamples were largely used in the context of explanation, asking questions and proof development, including induction, case splits and constructive arguments. They were also used in the context of errors being pointed out.

Examples were also suggested as useful cases, rather than directly supporting or refuting a given conjecture. For instance, worst case scenarios were brought up in the context of proof strategies.

There were no examples found being used in the context of rhetorical questions or requests for help, and very few in the context of directly considering the plausibility of a conjecture.

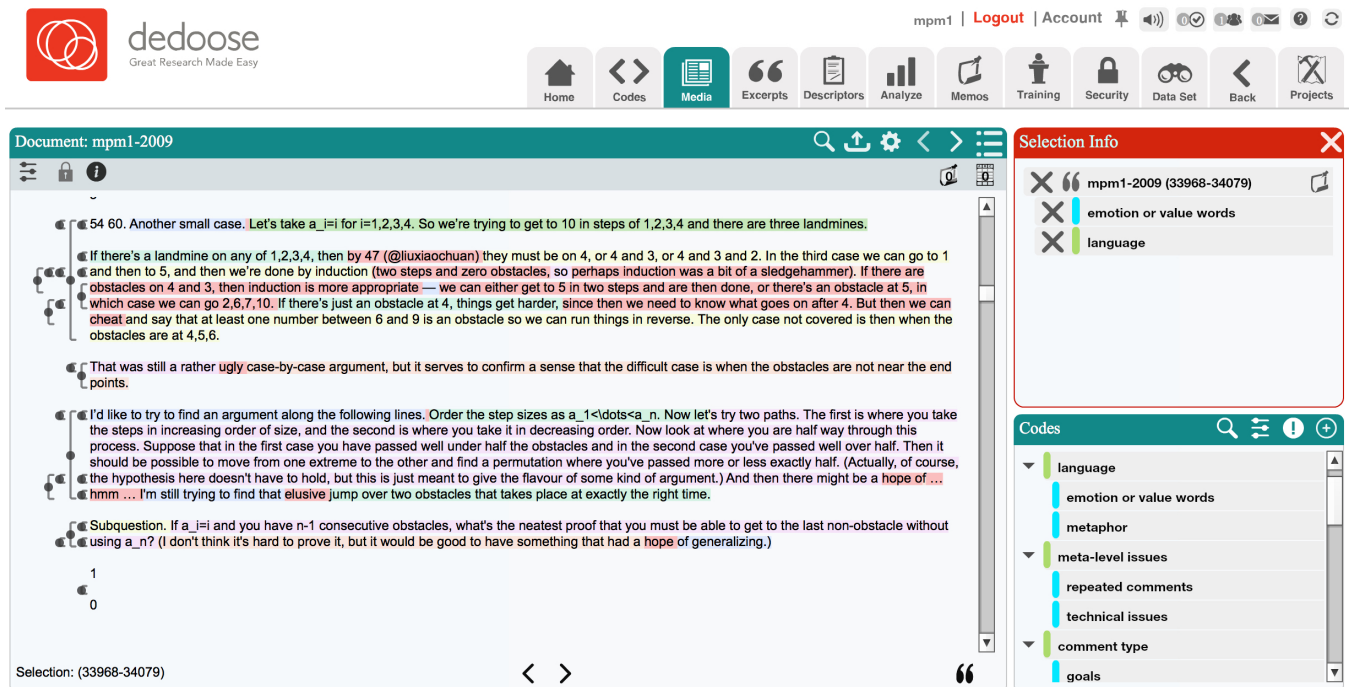


Figure 1: Example showing the codification of a comment using dedoose [2]

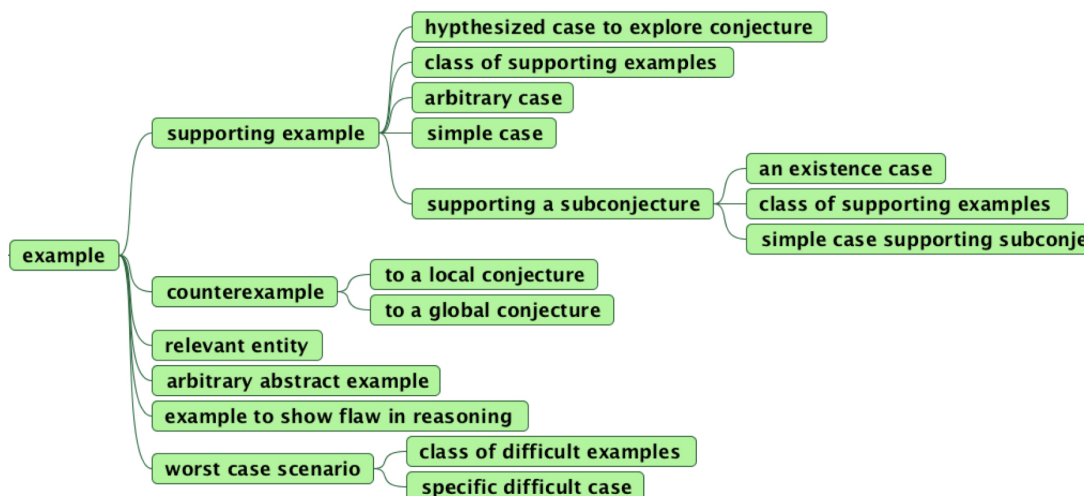


Figure 2: Interrelationships identified in Stages 2 and 3 between codes relevant to our examples category, where arcs represent subset relationships.

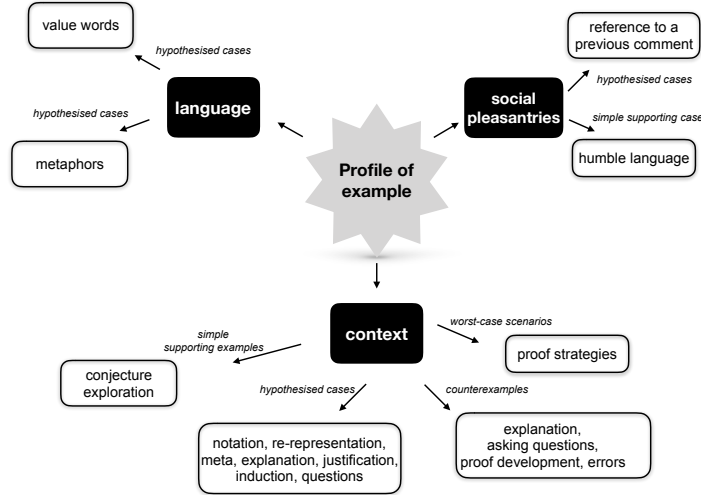


Figure 3: A diagrammatic model of our findings, showing the three central themes that emerged and how they were exhibited. Specific types of example in each case are shown in *italics*.

Language:

Value words were used describing some examples, for instance, examples were described as degenerate or as particularly useful (value words) in a given context. The quote below discusses an example, breaking it down into various scenarios and using them to discuss proof strategy and in particular when induction was “a bit of a sledgehammer” (metaphor), and when it was “more appropriate” (each time showing why). Emotive words such as “cheating” are used in a lighthearted way:

Let’s take $a_i = i$ for $i = 1, 2, 3, 4$. So we’re trying to get to 10 in steps of 1,2,3,4 and there are three landmines.

If there’s a landmine on any of 1,2,3,4, then by 47 (@liuxiaochuan) they must be on 4, or 4 and 3, or 4 and 3 and 2. In the third case we can go to 1 and then to 5, and then were done by induction (two steps and zero obstacles, so perhaps induction was a bit of a sledgehammer). If there are obstacles on 4 and 3, then induction is more appropriate – we can either get to 5 in two steps and are then done, or theres an obstacle at 5, in which case we can go 2,6,7,10. If theres just an obstacle at 4, things get harder, since then we need to know what goes on after 4. But then we can cheat and say that at least one number between 6 and 9 is an obstacle so we can run things in reverse. The only case not covered is then when the obstacles are at 4,5,6.

Social pleasantries:

When people were discussing a hypothesized case to explore, they frequently referred to a previous comment –

either by number, by the content, or using the name of the participant who had said it.

Humble language was used occasionally when exploring a simple case which supports a sub-conjecture: “(I may make mistakes here.)”

4 Further work and conclusions

The new records of mathematical reasoning, our ethnographic studies highlighting the importance of the example in such reasoning, alongside the development of sophisticated model generation systems, mean that we are now in a position to build on our insight into the use of examples in mathematics and extend it in a computational way. We plan to automate those roles (ii) and (iii) for which we found no corresponding system in §2, and to build on results from §3 to guide us in constructing a system which can useful contribute examples to a mathematics research conversation.

Building a system which can do even a limited aspect of “human-like” mathematics will open the way for a new form of mixed-initiative, collaborative reasoning between human and software participants in interactions which are both naturalistic but formally constrained and well-defined. This has the potential to impact both the professional practice and pedagogy of mathematics.

Acknowledgments

This work was supported by EPSRC grants EP/P017320/1 and EP/K040251/2. We are grateful to our anonymous reviewers for their thoughtful comments.

References

- [1] <http://events.inf.ed.ac.uk/sicsa-mcp/>.
- [2] Dedoose. www.dedoose.com.
- [3] Minipolymath3 project. <http://polymathprojects.org/2011/07/19/minipolymath3-project-2011-imo/>.
- [4] The polymath blog. <http://polymathprojects.org/>.
- [5] Mathoverflow. <http://mathoverflow.net>, September 2009.
- [6] A Bundy. Automated theorem provers: a practical tool? *Ann Math Artif Intell*, 61:3–14, 2011.
- [7] S. Colton. *Automated Theory Formation in Pure Mathematics*. Springer-Verlag, 2002.
- [8] S. Colton and A. Pease. The TM system for repairing non-theorems. In *Selected papers from the IJCAR'04 disproving workshop, Electronic Notes in Theoretical Computer Science*, volume 125(3). Elsevier, 2005.
- [9] T Hales et al. A revision of the proof of the kepler conjecture. *Discrete & Comp Geom*, 44, 2010.
- [10] G Gonthier. Advances in the formalization of the odd order theorem. *Proc. of Interactive Theorem Proving*, 6898, 2011.
- [11] T. Gowers and M. Nielsen. Massively collaborative mathematics. *Nature*, 461(7266):879–881, 2009.
- [12] W. T. Gowers. Rough structure and classification. *GAFA (Geometric And Functional Analysis)*, Special volume – GAFA2000(1–0), 2000.
- [13] J Harrison. andbook of practical logic and automated reasoning. 2009.
- [14] M. Johansson, L. Dixon, and A. Bundy. Conjecture synthesis for inductive theories. *Journal of Automated Reasoning*, 47(3):251–289, 2011.
- [15] I. Lakatos. *Proofs and Refutations*. Cambridge University Press, Cambridge, 1976.
- [16] Ursula Martin and Alison Pease. Mathematical practice, crowdsourcing, and social machines. In Jacques Carette, David Aspinall, Christoph Lange, Petr Sojka, and Wolfgang Windsteiger, editors, *Intelligent Computer Mathematics*, volume 7961 of *Lecture Notes in Computer Science*, pages 98–119. Springer Berlin Heidelberg, 2013.
- [17] W. McCune. MACE 2 reference manual. Technical Report ANL/MCS-TM-249,, Argonne National Laboratories, 2001.
- [18] W. McCune and R. Padmanabhan. *Automated Deduction in Equational Logic and Cubic Curves, LNAI, 1095*. Springer-Verlag, 1996.
- [19] W.W. McCune. Otter 3.0 Reference Manual and Guide. Technical Report ANL-94/6, Argonne National Laboratory, Argonne, USA, 1994.
- [20] M. Nielsen. *Reinventing Discovery: The New Era of Networked Science*. Princeton University Press, USA, 2011.
- [21] A. Pease. A computational model of lakatos-style reasoning. *Philosophy of Mathematics Education Journal*, ISSN 1465-2978 (Online)(27), April 2013.
- [22] A. Pease and U. Martin. Seventy four minutes of mathematics: An analysis of the third minipolymath project. In *Proceedings of the AISB Symposium on Mathematical Practice and Cognition II*, pages 19–29, 2012.
- [23] A. Pease and U. Martin. Summary of an ethnographic study of the third mini-polymath project. In *Computability in Europe*. Informal presentation, 2012.
- [24] G. Pólya. *Mathematics and plausible reasoning: Induction and analogy in mathematics*, volume I. Princeton University Press, 1954.
- [25] G. Pólya. *Mathematical Discovery*. John Wiley and Sons, New York, 1962.
- [26] A. Strauss and C. Juliet. Grounded theory methodology: An overview. In N. Denzin and Y. Lincoln, editors, *Handbook of Qualitative Research (1st ed)*, pages 273–284. Sage Publications, Thousand Oaks, CA., 1994.
- [27] H. Zhang and J. Zhang. Mace4 and sem: A comparison of finite model generators. In M. P. Bonacina and M. E. Stickel, editors, *Automated Reasoning and Mathematics: Essays in Memory of William W. McCune*, pages 101–130.